LOVELY PROFESSIONAL UNIVERSITY

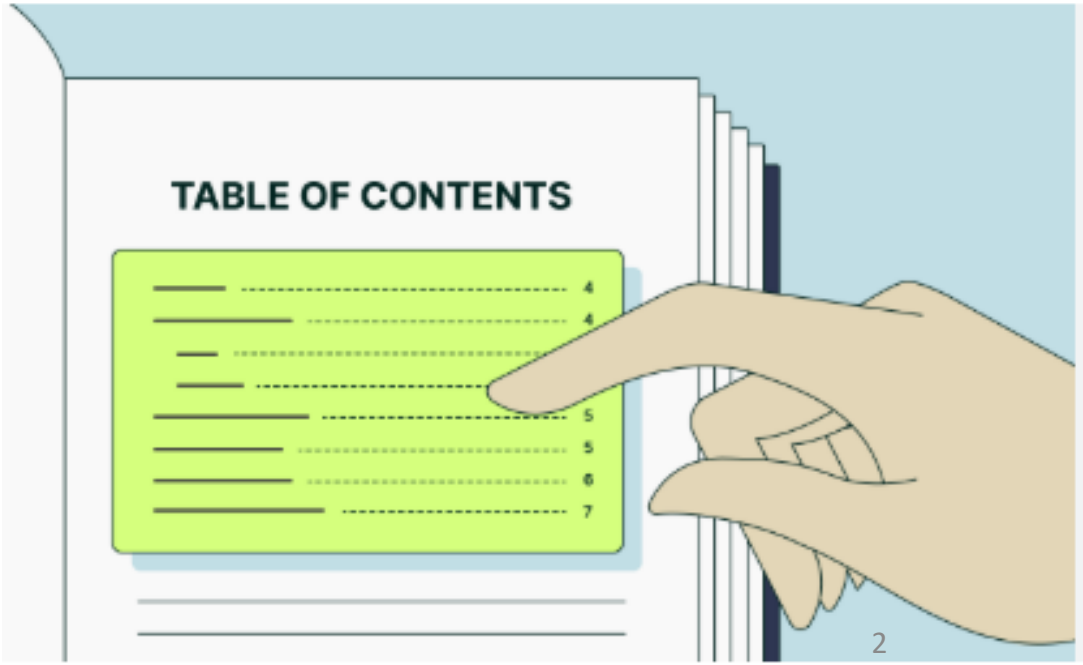
NAAC GRADE A++

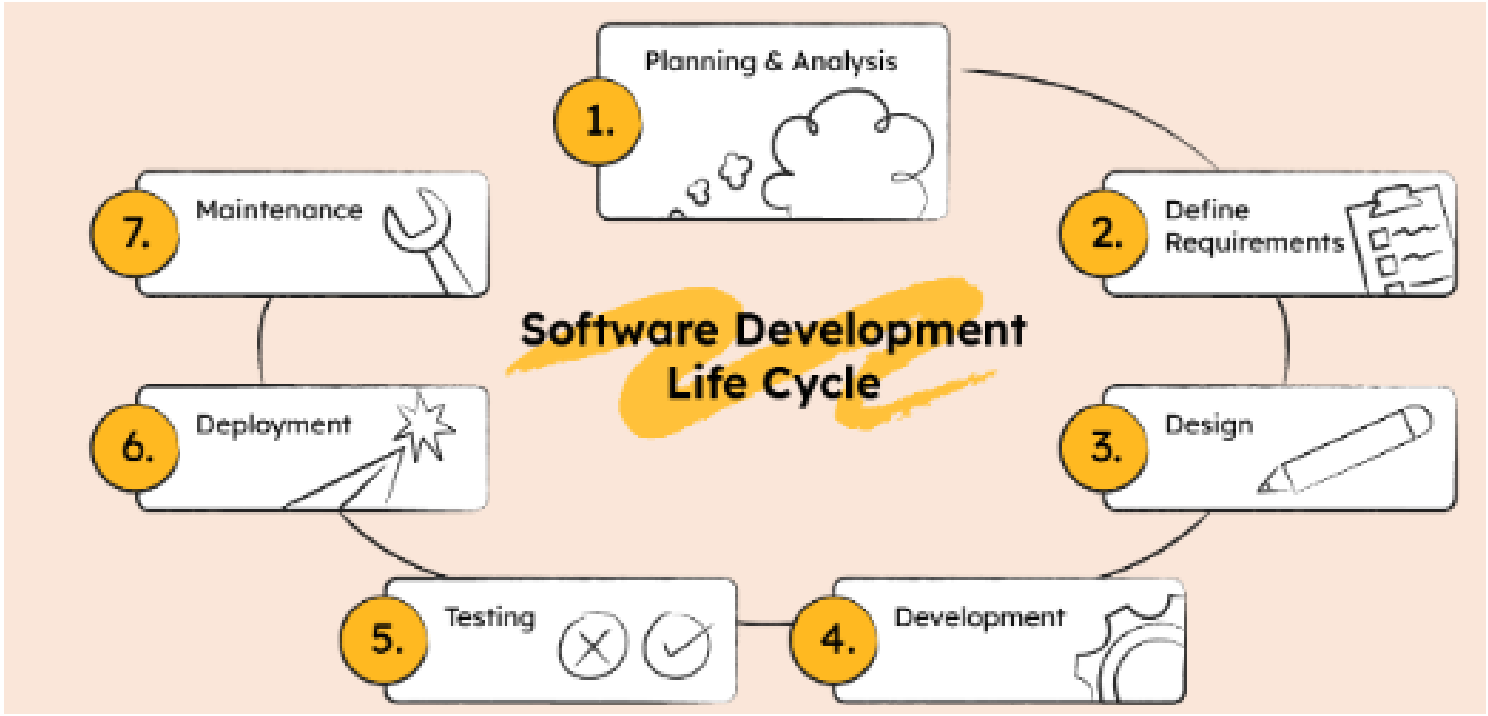
Unit-1

Introduction to S/w Engineering

Table of Content

- Software Life Cycle Models
- How SDLC works
- Types of Life Cycle model




Planning & Analysis
1.
2.
Define Requirements
3.
Design
4.
Development
5.
Testing
6.
Deployment
7.
Maintenance
Software Development Life Cycle

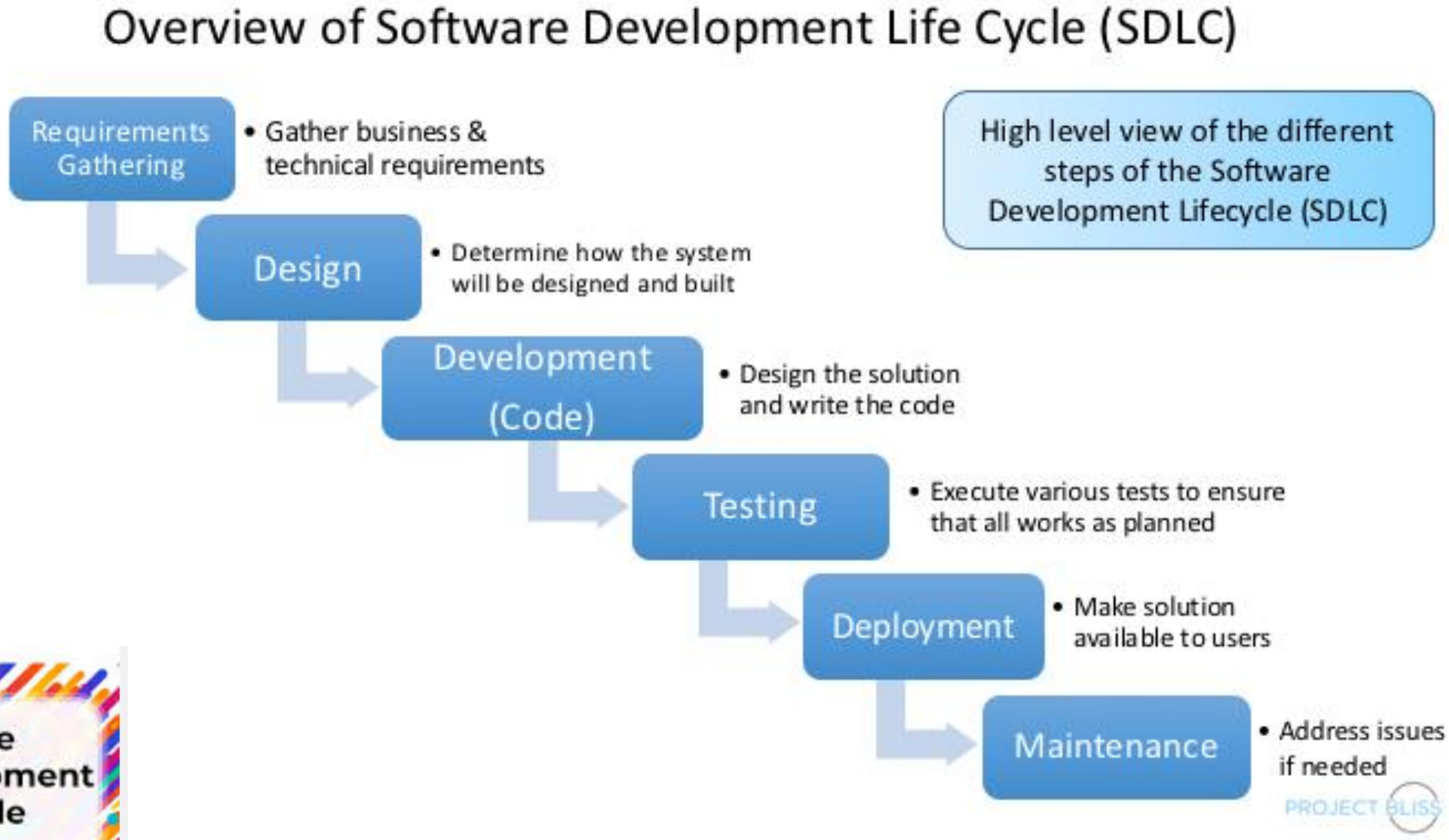

BENEFITS

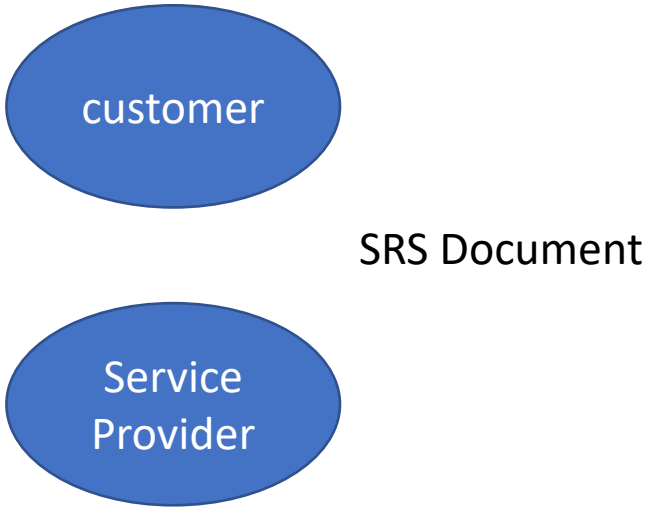

Benefits of a Strong SDLC
Better Overview of Development
Shorter Time-to-Market (TTM)
Maximized Productivity
Strategic Product Management
Enhanced Security
Realistic Estimates

Software Life Cycle Models or Process Model

- It is a pictorial and diagrammatic representation of the software life cycle.
- A life cycle model represents all the ***methods required to make a software product transit through its life cycle stages.*** It also captures the structure in which these methods are to be undertaken.
- SDLC provides a well-structured flow of phases
- That help an organization ***to quickly produce high-quality software*** which is well-tested and ready for production use

HOW SDLC WORKS?




customer
SRS Document
Service Provider

Contd.

Life-Cycle Models

- Life cycle models are **systematic approaches to understanding** the development and evolution of a system or project over time.

- These models provide a structured framework for planning, designing, implementing, testing, deploying, and maintaining systems

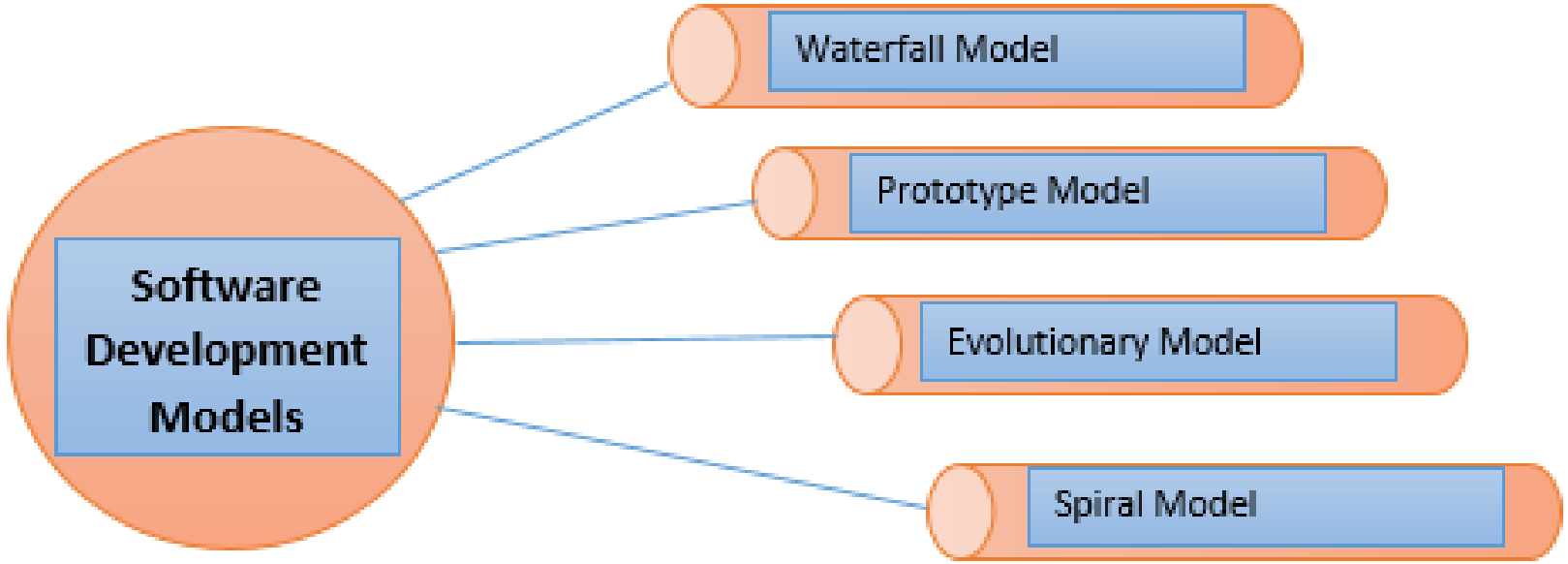
Contd.

Types of Life Cycle Model

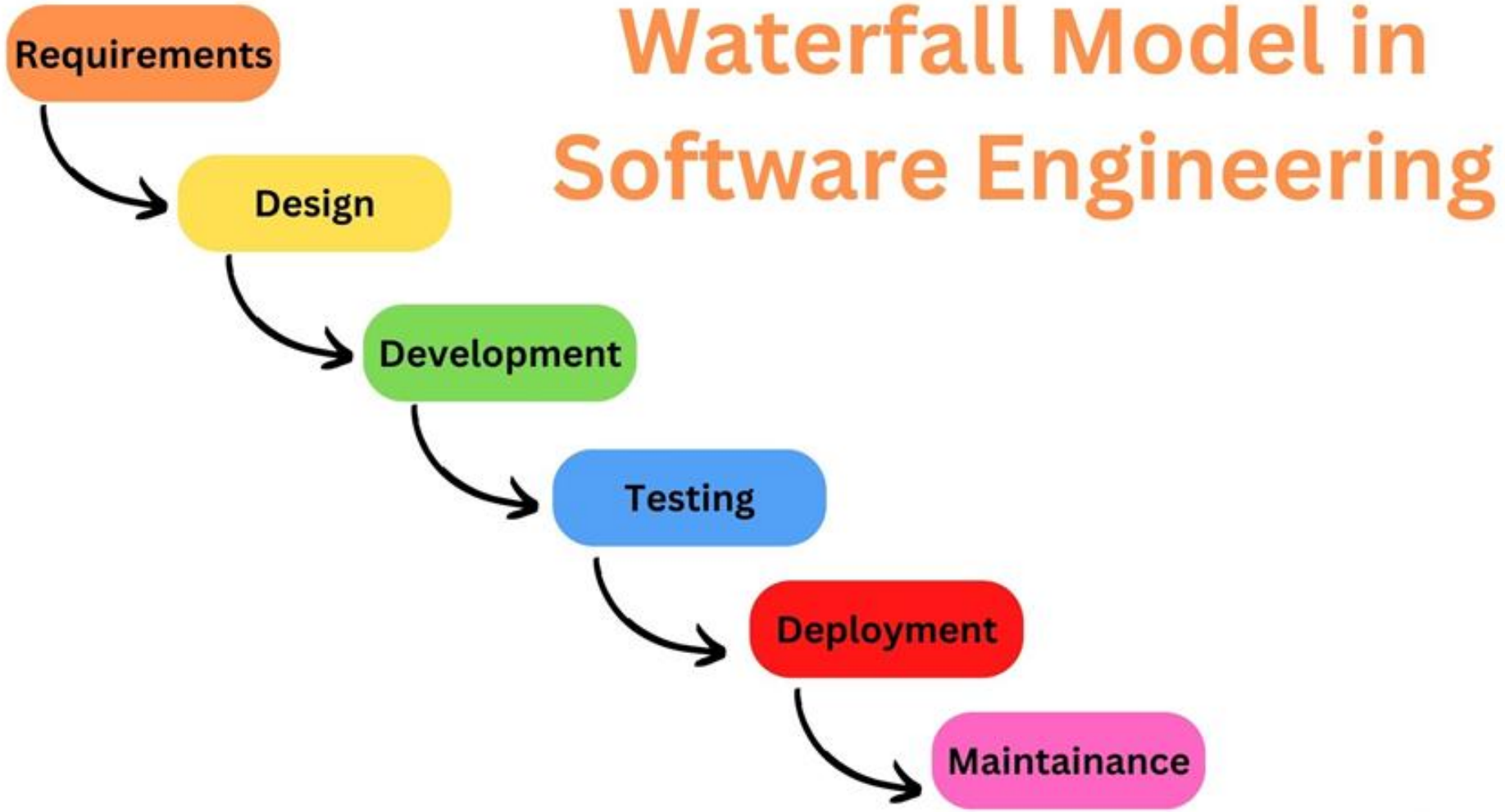
- Waterfall Model

Classical waterfall model
Iterative waterfall model

- Prototype Model
- Evolution and Spiral Model

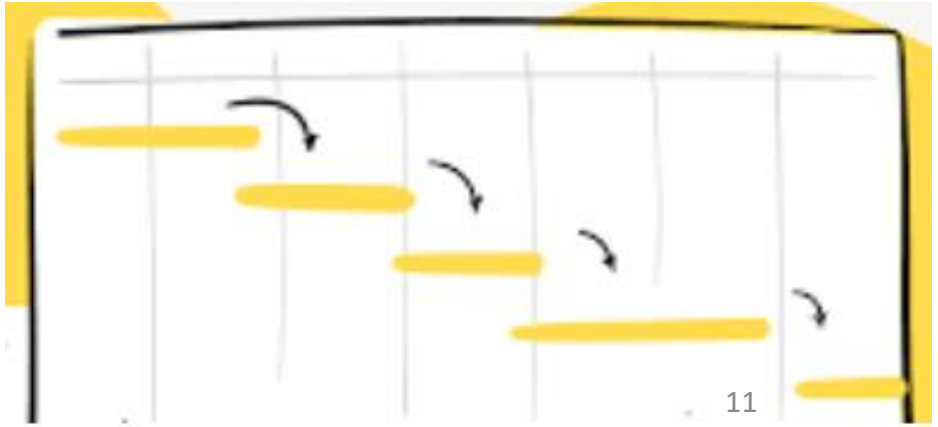


1. Waterfall Model



Contd.

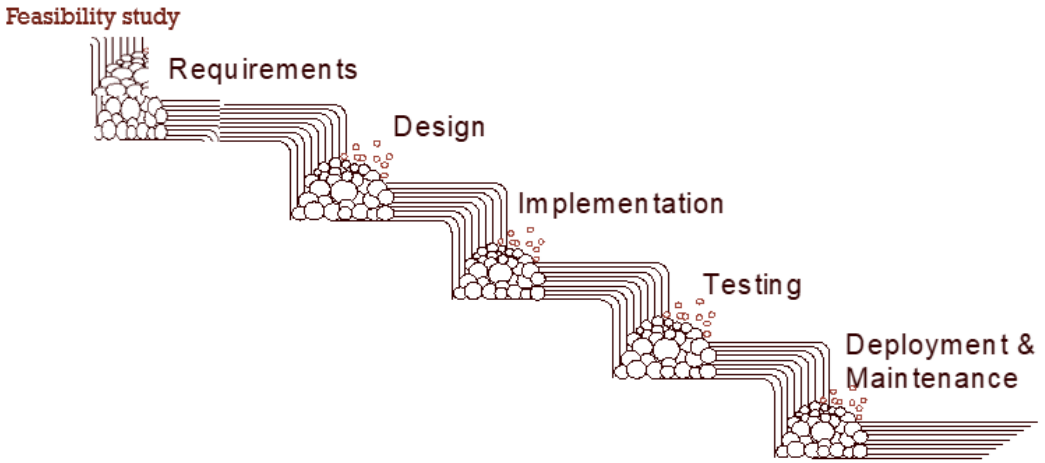
- **Linear and sequential approach to development**
- Progress flows steadily downwards like a waterfall
- **Oldest and most straightforward life cycle model**
- Progress flows in one direction through phases
- Each phase must be completed sequentially



Contd.

Initial Phase – Requirements Gathering

- **Interview stakeholders** to gather project requirements
- Document needs of clients and end-users
- Analyze requirements for clarity and completeness
- Ensure feasibility and project goals are established



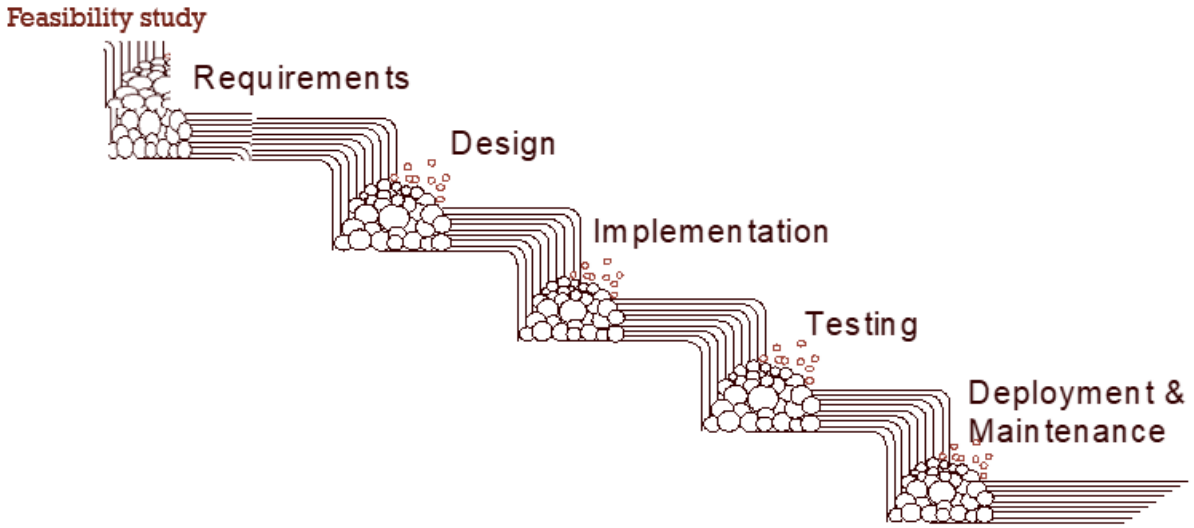
Next Phase – System Design Phase

- Create high-level system architecture and specifications
- Define system components and their relationships
- Develop a blueprint for subsequent phases

Contd.

Next Phase – Coding Phase

- Actual coding of software is performed
- Developers write code per design specifications
- Follow coding standards and best practices



Next Phase – Testing Phase

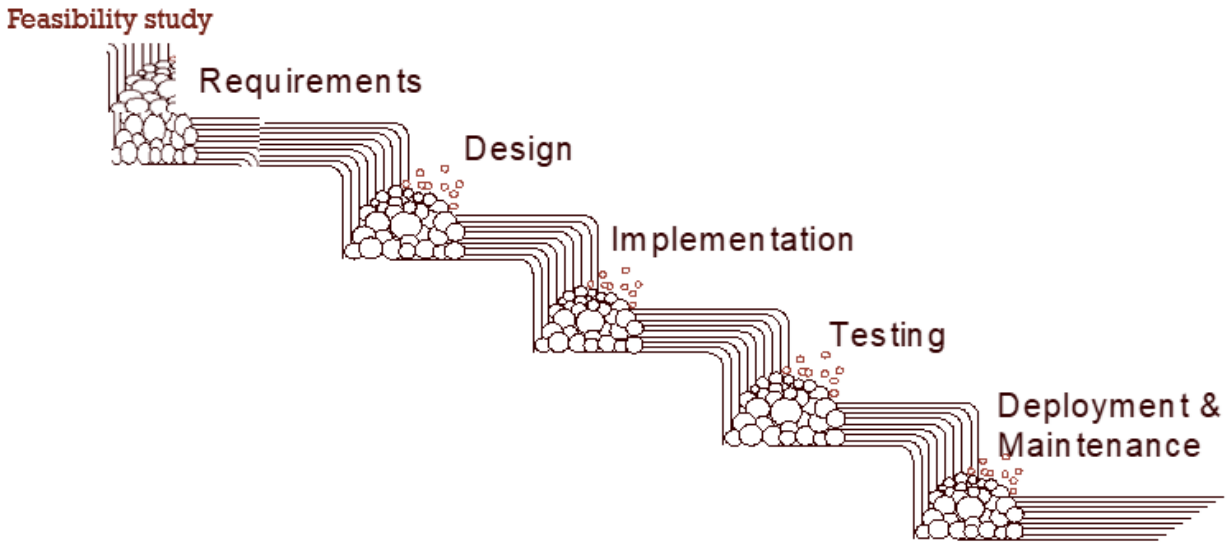
- System undergoes testing to detect defects
- Bugs are identified and fixed thoroughly
- Unit, integration, and system testing performed
- Ensures software meets specified requirements

Contd.

Next Phase – Deployment Phase

- Software is deployed to production environment
- End-users gain access to the software
- Activities include data migration and user training
- Necessary infrastructure is set up successfully

Next Phase – Maintenance Phase



Example of Waterfall-Model: Let's take the example of manufacturing a car

1. Requirements Phase:

Define specifications, model, features, and performance needs

2. Design Phase:

Create blueprints, schematics, and detailed component plans

3. Implementation Phase:

Manufacture car and assemble components sequentially

4. Testing Phase:

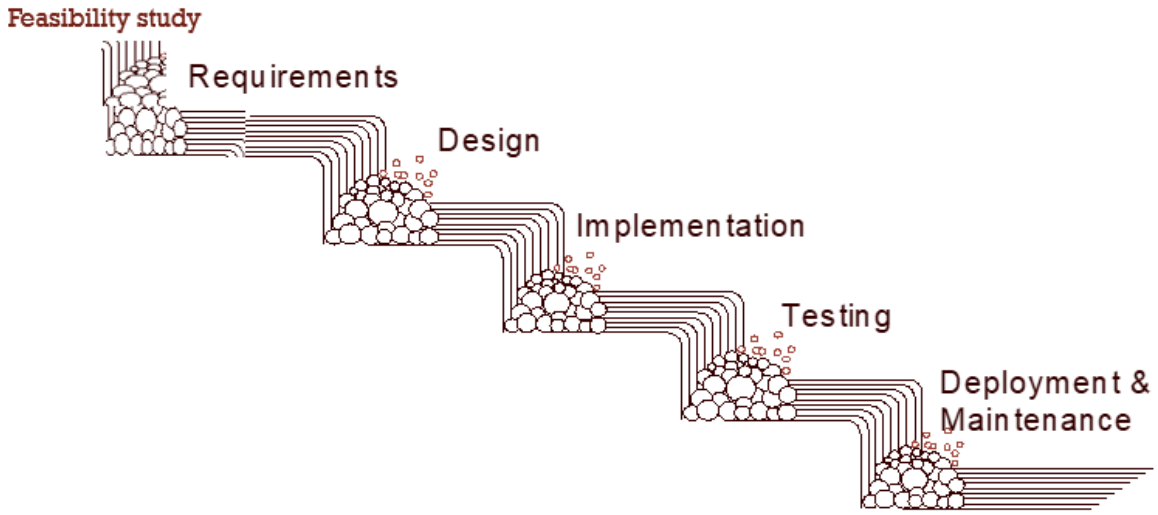
Perform performance, safety, and quality assurance **tests**

5. Deployment Phase:

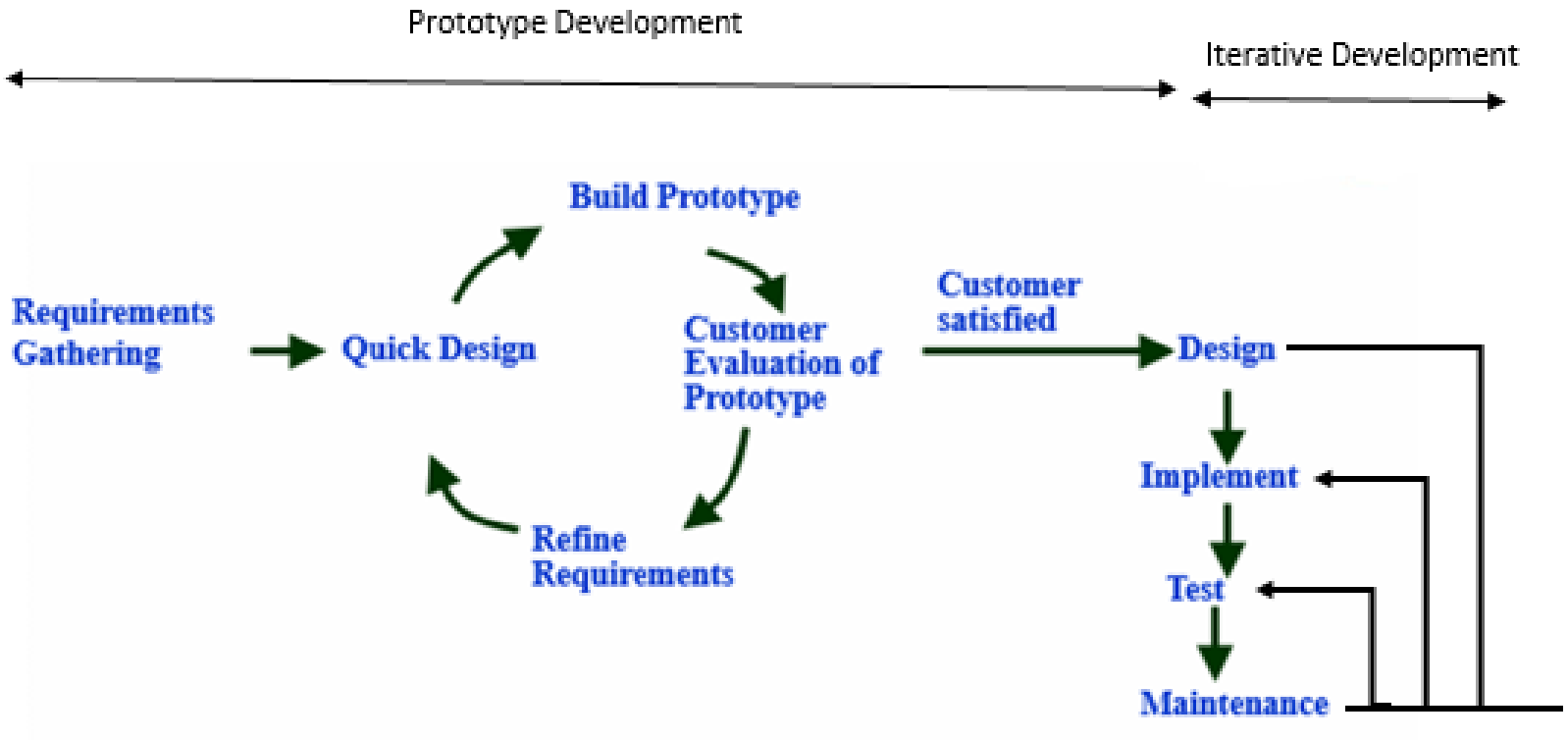
Deliver the product to customers or dealerships

6. Maintenance Phase:

Provide ongoing support, warranty services, and maintenance



2. Prototype Model



Prototype Model Phases

1. Requirements Gathering:

Collect initial requirements from stakeholders and users

2. Quick Design:

Develop a basic and preliminary design prototype

3. Prototype Development:

Create a working model for functionality demonstration

4. User Feedback:

Gather feedback on the prototype for improvements

5. Refinement:

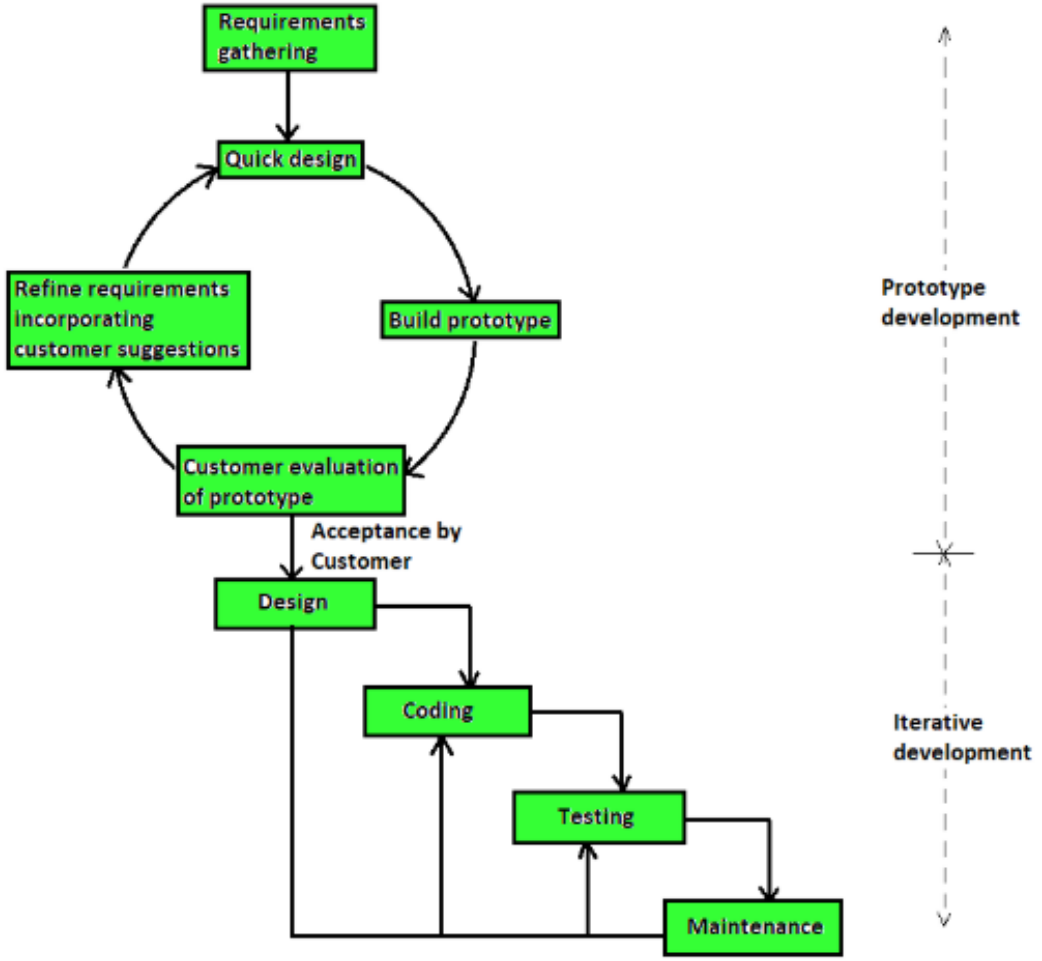
Refine the prototype based on user input and analysis

6. Final Product:

Develop the actual product using improved specifications

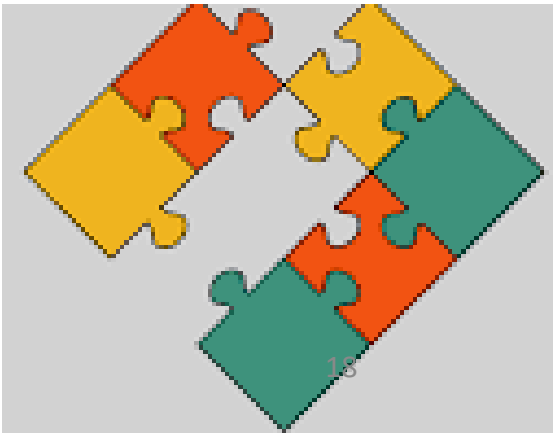
Prototype Model Overview

- **Reason for Prototype:** Impossible to “get it right” initially
- **Build Before Development:** A working system prototype is created
- **Prototype Characteristics:** Limited functions, low reliability, inefficient performance
- **Best for Unclear Requirements:** Ideal when customers lack clear ideas
- **Throw-Away Model:** Used for risk mitigation in technical/requirement risks
- **Customer Involvement:** Clients give feedback during system design
- **Refine Requirements:** Feedback refines the prototype until user approval
- **Iterative Cycle:** Repeats between developers and customers for improvements



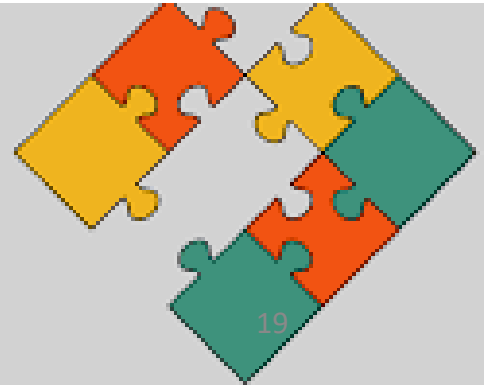
Contd. Advantages:

- **Better Requirement Understanding:** Clarifies user needs early in development
- **Improved User Involvement:** Client provides continuous feedback
- **Reduced Risk:** Early detection of technical and requirement issues
- **Enhanced System Design:** Allows iterative improvements based on feedback
- **Saves Time:** Helps identify flaws early, avoiding late-stage changes



Contd. Disadvantages:

- **Higher Development Costs:** Iterative changes increase project expenses
- **Increased Time:** Building prototypes delays actual development
- **Incomplete Solutions:** Final product may differ from prototype
- **User Confusion:** Users may mistake the prototype for the final system
- **Inefficient Performance:** Prototype may lack reliability and optimization

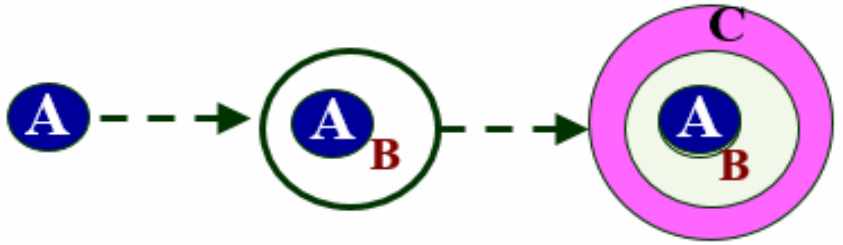


Difference

The **Waterfall model** follows a linear and sequential approach,
whereas the **Prototype model** emphasizes iterative development through
building and refining prototypes before the final system is developed.

Evolutionary Model

Many successive iterations/versions are implemented and delivered to the customer until the desired system is released

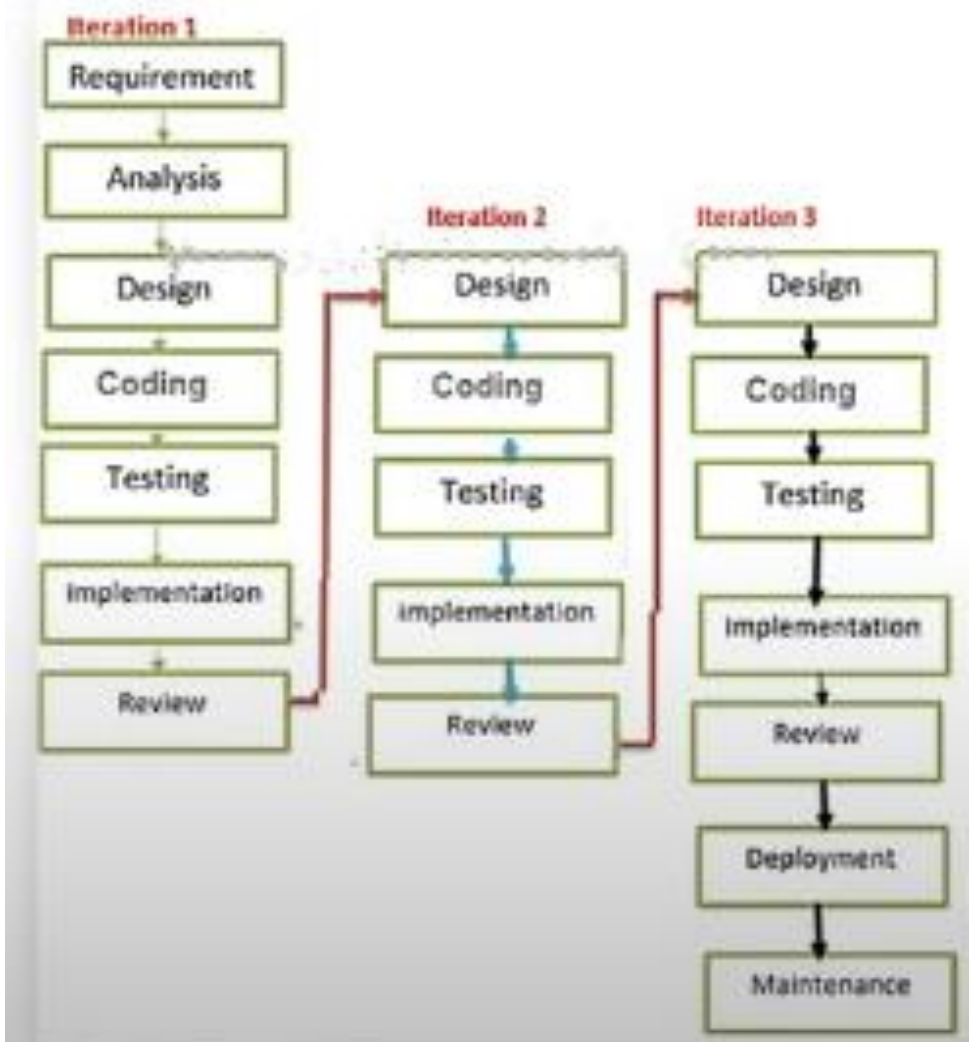


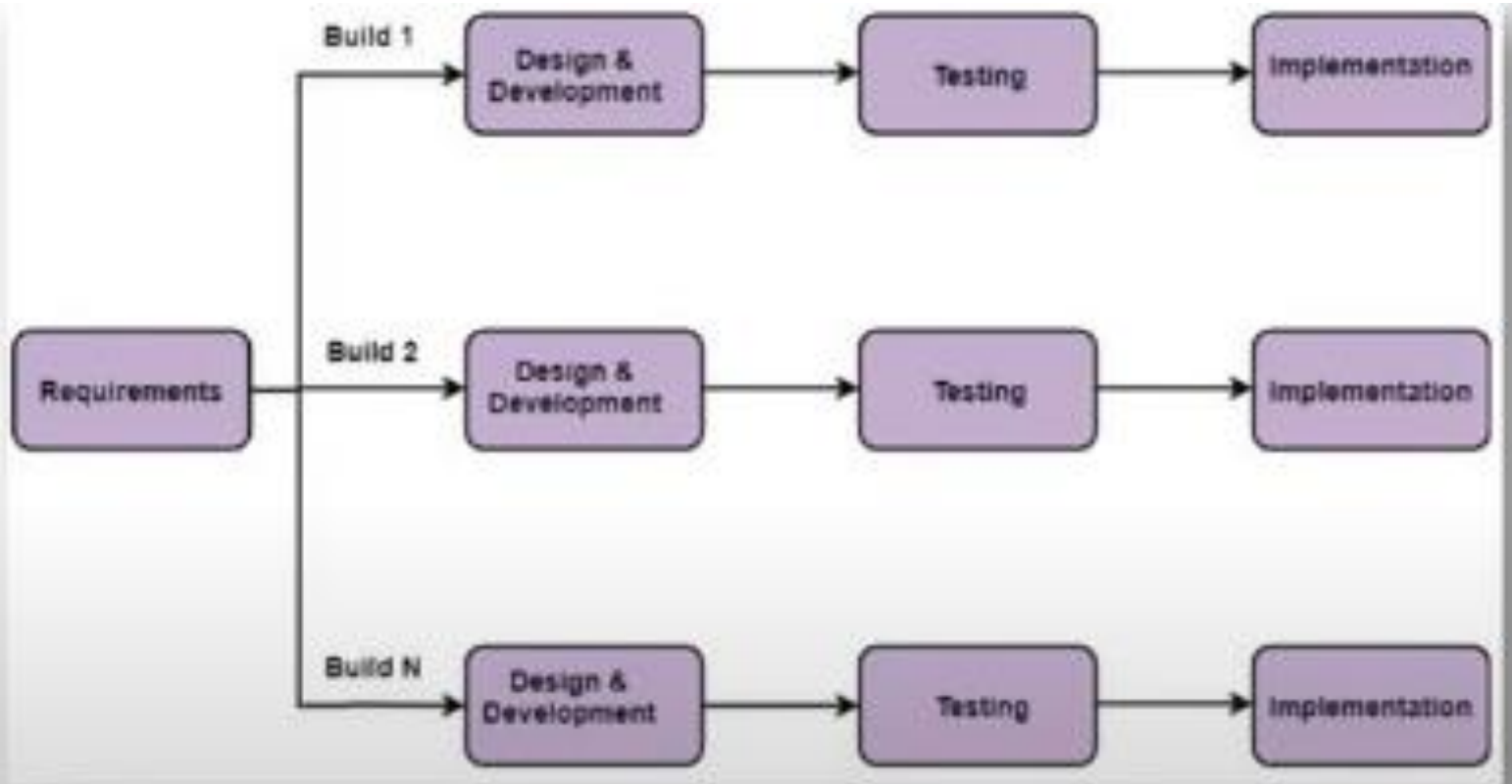
- Evolutionary model is also referred to as the ***successive versions model.***
- It is the **combination of iterative and incremental model**.
- Known as ***Design a little, build a little, test a little and deploy a little***.
- Incremental model first implement a few basic features and deliver to the customer. Then build the next part and deliver it again and repeat this step until the desired system is fully realized. No long term plans are made here.

  E.g. metro system

- Iterative model main advantage is its feedback process in every phase.

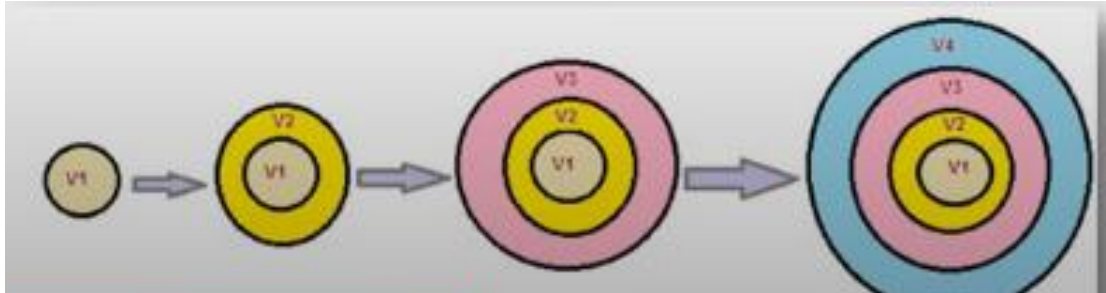
Revision of Iterative Model and Incremental Model





When to used Evolutionary Process Model ?

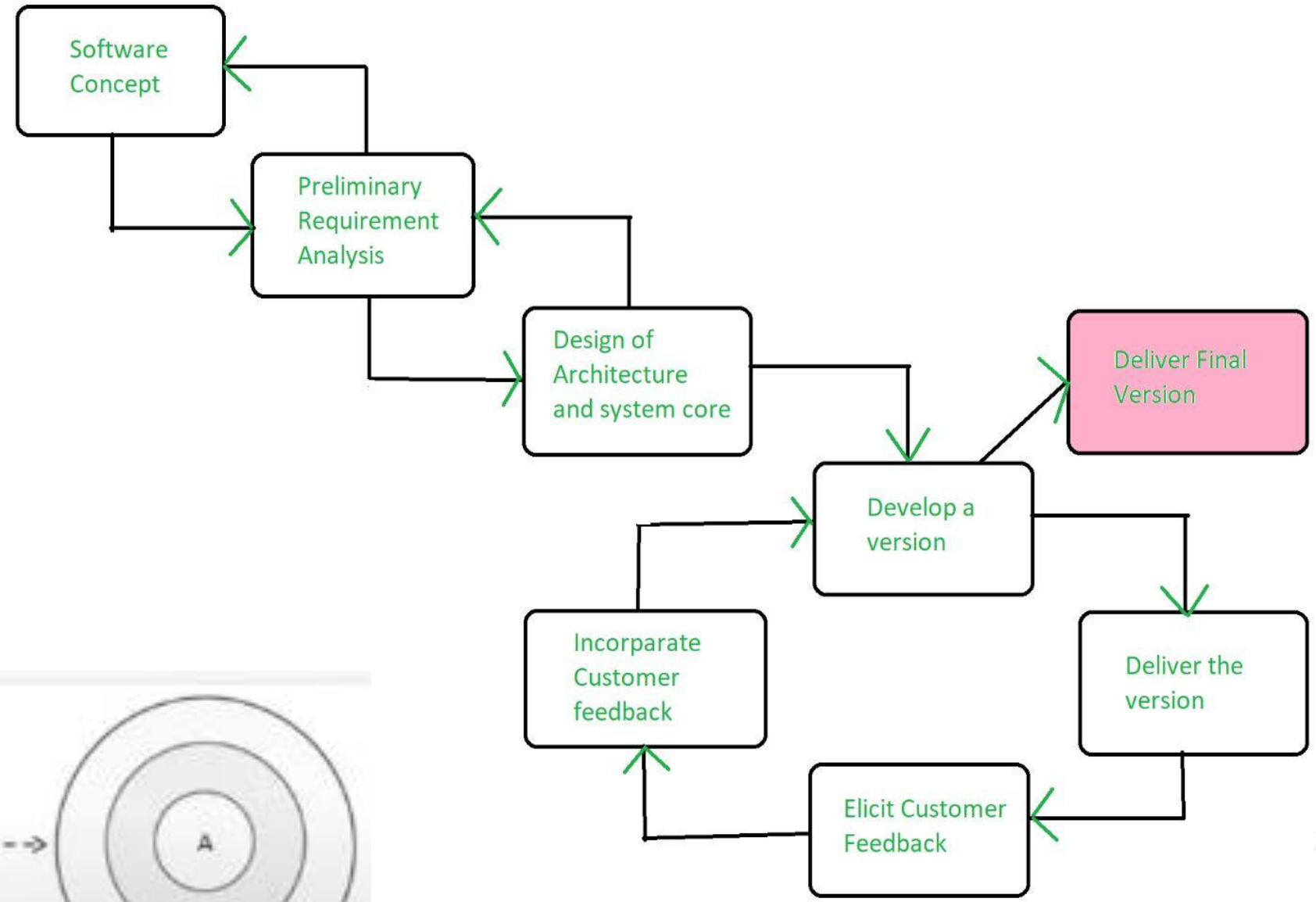
- Used ***in large projects*** (step by step development of large project)
- Used ***when the customer wants to start using the core features*** instead of waiting for the full software
- Customer requirement are ***not fixed. But clear*** concept.
- Small changes required in separate module
- (Delivers work chunks to customers incrementally).



S/w requirement is first ***broken down into several modules***

That can be incrementally constructed means it **take customer feedback of each module**

Delivered product module by module to the customer



How the **Evolutionary Model** can be explained using the **Delhi Metro Project**:

1. Software Concept (Metro Concept):
The concept was to build a metro system to reduce traffic congestion and improve public transportation in Delhi.

2. Preliminary Requirement Analysis:
Initial requirements were gathered, such as routes, number of stations, and feasibility studies for connecting key areas in Delhi.

3. Design of Architecture and System Core:
The core infrastructure was planned, such as tracks, signalling systems, stations, and the selection of rolling stock (trains).

4. Develop a Version (Phase Development):
The metro construction was divided into **phases**:

Phase 1: Focused on building essential metro lines like the Red Line, Blue Line, and Yellow Line to cover the major routes.

5. Deliver the Version:
After the completion of Phase 1, these metro lines were opened to the public for use.

6. Elicit Customer Feedback:
Public feedback was collected on the service, including the **need for better connectivity, frequency of trains**, and additional ***features like smart card systems.***

7. Incorporate Customer Feedback:
Feedback led to improvements like **increasing train frequency, adding more metro coaches**, and enhancing facilities like **escalators a**nd lifts.

8. Repeat Development Cycle (New Phases):

- **Phase 2**: Expanded the network with more lines to cover additional areas.
- **Phase 3**: Introduced advanced technologies, better train systems, and extended the metro network even further.
- **Phase 4**: Planned for future expansion to new areas based on demand and feedback.

Contd.

- **Deliver Final Version**:
  Although the metro continues to evolve, with each phase, a more complete and efficient version of the Delhi Metro is delivered to the public.

Key Points in the Evolutionary Model for Delhi Metro:

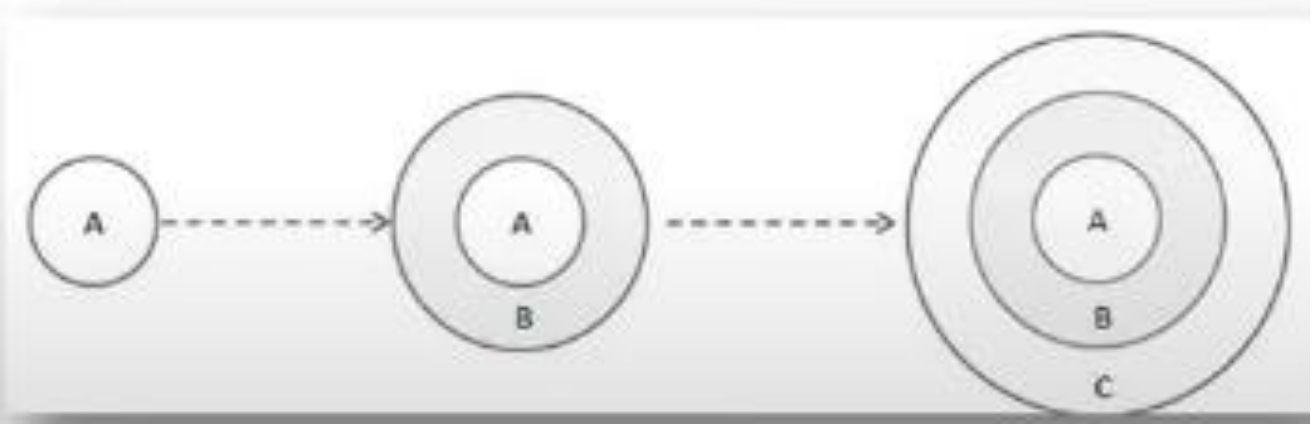
- The **core infrastructure** (tracks, trains, signalling) was developed first to ensure basic usability.
- **Incremental delivery** allowed the public to use parts of the system while further expansions were underway.
- **Feedback loops** helped in identifying issues like overcrowding or the need for more connectivity.
- **Adaptability** enabled the inclusion of new technologies like contactless cards and automatic train operations in later phases.

Outcome:

- The **Delhi Metro** evolved iteratively, improving and expanding over time, just like in the **Evolutionary Model**, delivering a functional and efficient system at every stage while continuously enhancing it based on user needs and feedback.

Evolutionary Model

- Combines Iterative and Incremental development models.
- Delivers system in smaller, incremental releases.
- Initial requirements and architecture are envisioned early.
- Adapts features based on user feedback.
- Focuses on evolving software through development cycles.



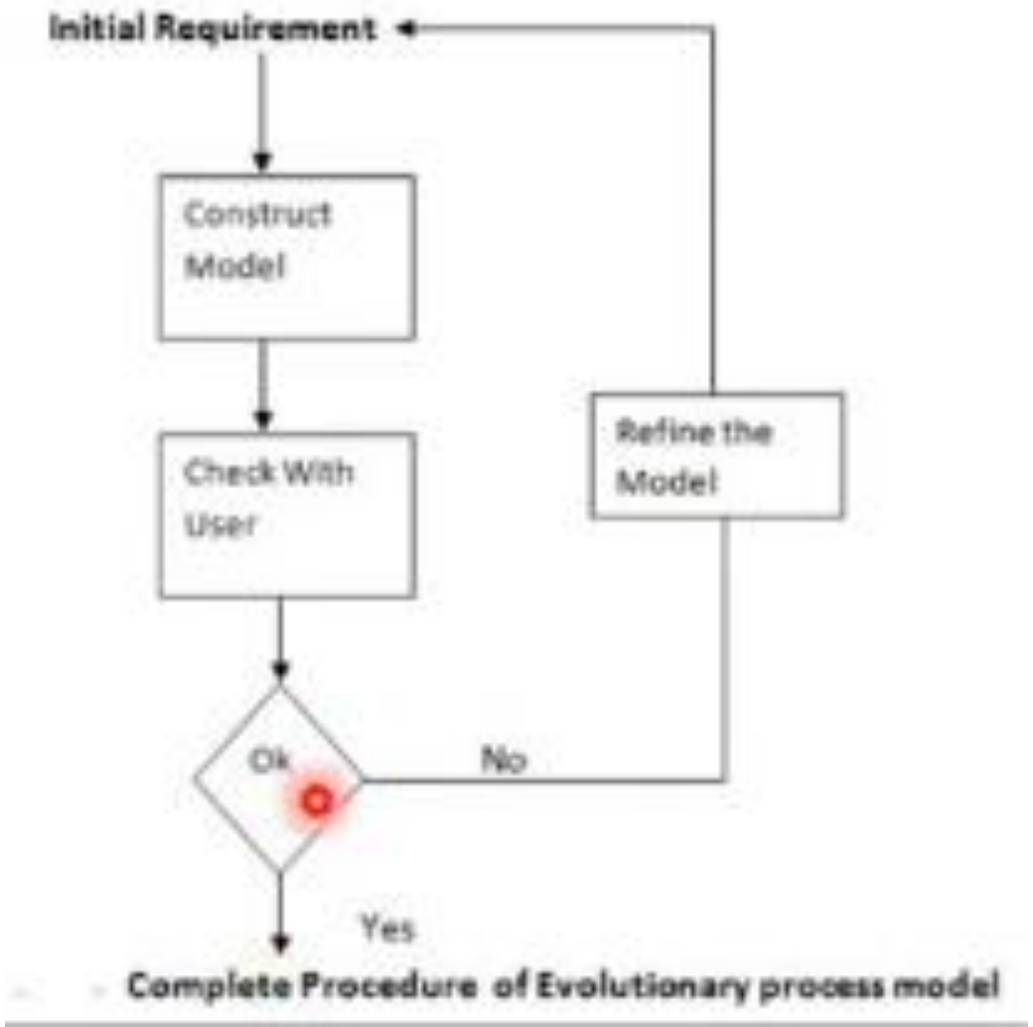
Incremental Approach Iterative Approach

Characteristics of Evolutionary Model

- Reduces delivery time with iterative cycles.
- Breaks work into smaller, prioritized chunks.
- Delivers work chunks to customers incrementally.
- Allows frequent validation of customer requirements.
- Accommodates changing requirements throughout development.

Necessary Conditions for Evolutionary Model

- Customer needs must be clear and well-explained.
- Small changes allowed, but not major modifications.
- Adequate time must exist for market constraints.
- High risk requires continuous reporting to customers.
- Suitable for projects involving new, unfamiliar technology.



Example of Evolutionary Model

- **E-commerce Platforms**: Amazon expand features iteratively.

Core features like shopping cart delivered first; advanced features like recommendations and AI added later.

- **Mobile Apps**: WhatsApp adds features based on user feedback.

WhatsApp launched with basic messaging; new features like calls, payments added incrementally.

- **Social Media Platforms**: Facebook evolves via incremental updates.

Facebook started as a basic social network; features like marketplace and stories evolved over time.

- **Game Development**: Games released with core features, updated later.

Games like Fortnite **launch with essential gameplay**; ***frequent updates*** bring new maps and modes.

Contd.

- **Operating Systems**: Windows, Linux improve over time with iterations.

Windows and Linux release initial versions; upgrades bring new features and fixes incrementally.

- **AI Systems**: ChatGPT evolves using user feedback and refinements

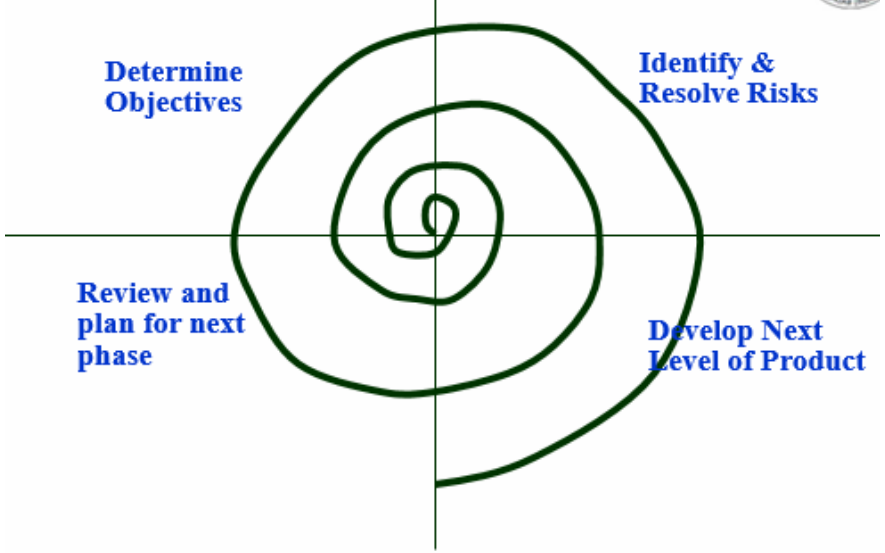
ChatGPT launched ***with basic conversational abilities***; updated based on ***user feedback and testing***.

❖The **spiral model** emphasizes risk management and is ideal for large, high-risk projects, **combining iterative development with formal evaluation phases.**

❖ The **evolutionary model** focuses on ***incremental delivery*** of working software, adapting to changing requirements through simpler, faster iterations.

4. Spiral Model

- Best model ***for risky***, complex, and extensive projects.
- **Visualized** as a spiral with iterative loops.
- **Loop count** depends on ***project type*** and requirements.
- Combines iterative and waterfall model elements effectively.
- Starts with small requirements and iterative development phases.
- Adds functionality at any spiral phase incrementally.



- Risk Handling
- **Developed by** Barry Bohem,1986.
- Main **purpose is to reduce the risk.**
- ***Radius*** of spiral represents **cost**
- **Angular dimension** represents **progress**, as the **degree of angle increases**
- **Known as Meta model** (because it uses multiple models)

  Step by step → waterfall model

  Feedback → iterative model

  Prototype → due to risk

- Software is developed in a series of incremental releases as per each spiral.

E.g. gaming industry, Microsoft OS versions.

Advantages

- Risk Handling (Risk are analysed at the early stage of project development)
- Large projects
- Flexible
- Customer Satisfaction

Disadvantages

- Expensive
- Too much Risk analysis
- Time

When to use Spiral Model ?

- When the project is large and high budget
- When requirements are unclear and complex
- Where the software needs continuous risk evaluation
- When creation of prototype is applicable
- When changes may be require at any time

Contd. Four key phases for software development

1. Planning (Requirement Gathering & analysis)

- Communication between customers and project head.
- Collects, understands, and identifies core objectives and solutions.
- Gathers detailed requirements and alternative solutions iteratively.

2. Risk Analysis (Risk Management)

- Identifies potential risks during development phases.
- Develops strategies to mitigate and manage risks effectively.
- Design a risk-free prototype

Contd.

3. Engineering & Execution (Design and Coding)

- Designs software using prototypes as architectural guidelines.
- Implements coding, testing, and initial deployment processes.
- ***Gathers client feedback during deployment*** for future improvements**.**

4. Customer Evaluation (Review and Deployme

- Reviews client feedback, observations, and required changes.
- Fixes issues and creates a new baseline spiral.
- Final product is launched after all corrections are addressed.

Software development for an online banking system

- **Plan Phase (Objective Setting):**
  - **Example:** In developing an online banking system, the first phase involves setting goals like secure login, transaction processing, and customer account management.
  - **Activities:**
    - Define functional requirements (e.g., account creation, transaction limits).
    - Set timelines and budgets.
    - Identify the target user base and security needs.

- **Risk Analysis Phase:**
  - **Example:** Identify risks related to security breaches, data loss, and system downtime.
  - **Activities:**
    - Simulate security threats (e.g., hacking attempts).
    - Assess software vulnerabilities and potential failure points.
    - Develop strategies for risk mitigation, like encryption and multi-factor authentication.

- **Engineering Phase (Development & Testing):**
  - **Example:** Begin developing the online banking platform.
  - **Activities:**
    - Design the system architecture, user interfaces, and databases.
    - Develop the login system, transaction functionality, and integration with external services.
    - Conduct unit tests and integration tests to ensure system functionality.
- **Iterative Refinements in Subsequent Spirals**
  - **2nd Iteration:** Address feedback from early testing, enhance the user interface, and improve security measures.
  - **3rd Iteration:** Introduce advanced features such as AI-driven fraud detection and mobile app integration.
  - **Final Iteration:** Deploy the system in a real-world environment, test it with real users, and ensure it meets regulatory and security standards.

Contd.

Advantages

- Best model for **managing risky software development projects.**
- Suitable for **large, complex software development** initiatives.
- Ensures better customer satisfaction with early client involvement.
- Follows iterative and incremental approaches for effective development.
- Open to receiving and incorporating regular feedback effectively.
- Delivers improved quality through multiple development phase repetitions.
- Flexible with requirements at any phase or development stage.

Contd.

Disadvantages

- More complex compared to other SDLC methods.
- Requires higher budget; **not suitable** for small or medium projects.
- Highly dependent on expert developers and risk analysts.
- Complicates time tracking with unknown number of phases.
- Multiple iterations can be tedious(lasting for long time) for development teams.
- Time-consuming due to frequent changes and evaluations.
- Requires intensive investment for planning and risk analysis

Difference

Aspect	Spiral Model	Evolutionary Model
Focus	Risk management	Incremental development
Iteration	Formal spirals	Simple, incremental iterations
Risk Assessment	Explicit and prioritized	Not a primary focus
User Feedback	At the end of each spiral	Continuously in iterations
Complexity	High	Moderate
Best For	Large, high-risk projects	Dynamic, evolving projects
Cost	Higher	Lower
Time to Deliver	Longer	Quicker

MCQ

- 1. A model that is the demo implementation of the system.
  a) waterfall
  b) prototype
  c) incremental
  d) agile
- 2. Maintenance is the final phase in waterfall model.
  a) True
  b) False
- 3. A stage in which individual components are integrated and ensured that they are error-free to meet customer requirements.
  a) Coding
  b) Testing
  c) Design
  d) Implementation

- 4. _____ is a step in which design is translated into machine-readable form.
  a) Design
  b) Conversion
  c) Debugging
  d) Coding
- 5.What do you call a technical person who is capable of understanding the basic requirements?
  a) team leader
  b) analyst
  c) engineer
  d) stakeholder
- 6. A step in waterfall model that involves a meeting with the customer to understand the requirements.
  a) Requirement Gathering
  b) SRS
  c) Implementation
  d) Customer review

- Methodology in which project management processes were step-by step.
  a) Incremental
  b) Waterfall
  c) Spiral
  d) Prototyping
- A planned program if work that requires a definitive amount of time, effort and planning to complete.
  a) Problem
  b) Project
  c) Process
  d) Program
- Processes for evolving a software product depend on:
  a) Type of software to be maintained
  b) Development processes used
  c) Skills and experience of the people involved
  d) All of the mentioned

- Spiral model was **developed by**
  a) Victor Bisili
  b) Berry Boehm
  c) Bev Littlewood
  d) Roger Pressman

- Software evolution **does not** comprises:
  a) Development activities
  b) Negotiating with client
  c) Maintenance activities
  d) Re-engineering activities

- **What is the main focus of the evolutionary model?**
  a) Risk management
  b) Building a working version of the software
  c) Cost reduction
  d) Testing of software
- **Which of the following is a key feature of the evolutionary model?**
  a) Linear sequence of activities
  b) Delivery of incremental versions of the software
  c) Focus on documentation over coding
  d) Single release of the final product
- **The evolutionary model is best suited for:**
  a) Large projects with well-defined requirements
  b) Projects with rapidly changing requirements
  c) Projects with a fixed budget and timeline
  d) Safety-critical systems

- **In the evolutionary model, user feedback is collected:**
  a) Only at the end of the project
  b) At each iteration
  c) During the initial planning stage only
  d) After deployment

Spiral Model MCQs

- **The spiral model combines features of which two models?**
  a) Waterfall and Agile
  b) Prototype and Incremental
  c) Prototype and Waterfall
  d) Waterfall and RAD
- **What is the primary focus of the spiral model?**
  a) Project management
  b) Risk analysis
  c) Cost estimation
  d) Documentation

- **How many phases does each loop in the spiral model consist of?**
  a) 2
  b) 3
  c) 4
  d) 5

- **Which of the following is NOT a characteristic of the spiral model?**
  a) Cyclic process
  b) Risk-driven approach
  c) Linear progression
  d) Iterative development

- **In the spiral model, risk is analyzed during which phase?**
  a) Planning
  b) Risk assessment
  c) Engineering
  d) Evaluation

- **Who proposed the spiral model?**
  a) Barry Boehm
  b) Fred Brooks
  c) Winston Royce
  d) James Martin

- **The spiral model is most appropriate for:**
  a) Small projects with simple requirements
  b) Projects with high risks and changing requirements
  c) Static, well-defined projects
  d) Fixed-cost projects

- **Which of the following is a drawback of the spiral model?**
  a) High flexibility
  b) High cost and complexity
  c) Focus on risk management
  d) Incremental delivery

- **The four main phases of the spiral model are:**
  a) Planning, Risk Analysis, Development, Testing
  b) Planning, Risk Analysis, Engineering, Evaluation
  c) Requirement, Design, Coding, Testing
  d) Analysis, Development, Testing, Maintenance
- **In the spiral model, the amount of work done per iteration:**
  a) Increases with time
  b) Decreases with time
  c) Remains constant
  d) Is unpredictable